



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/904,882	07/12/2001	Calvin White	40101/02401	3600

7590 06/30/2004
FAY KAPLUN & MARCIN, LLP
150 BROADWAY
SUITE 702
NEW YORK, NY 10038

EXAMINER

VO, TED T

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 06/30/2004

8

Please find below and/or attached an Office communication concerning this application or proceeding.

1924

Office Action Summary	Application No. 09/904,882	Applicant(s) WHITE ET AL.	
	Examiner Ted T. Vo	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on 09 March 2004.

2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) 1-12 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) 1-12 is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) ☐ All b) ☐ Some * c) ☐ None of:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. _____.

3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s)/Mail Date. _____
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____	6) <input type="checkbox"/> Other: _____

U.S. Patent and Trademark Office
PTOL-326 (Rev. 1-04)

Office Action Summary

Part of Paper No./Mail Date 8

DETAILED ACTION

1. This action is in response to the Amendment filed on 03/09/2004.
Claims 1 and 4 are amended. Claims 1-12 are pending in the application.
Claims 1-12 stand finally rejected as being anticipated by Abbott.

Response to Arguments

2. With regard to the amendment of Claims 1 and 4, the rejection of Claims 1-7 under 35 U.S.C 101 is withdrawn.

3. Applicants' arguments in Remarks (pages 7-9) with respect to Claims 1-12, which are rejected under 35 U.S.C. 102(e) as being anticipated by Abbott, have been fully considered. However, the arguments are not persuasive.

With regard to Claim 1, Applicants argue that Abbott reference merely discloses a system for locating native libraries (Remarks: page 7, lines 1-2 of the last paragraph) and not loading classes from a plurality of stream sources where Applicants contend that there is no teaching or suggestion in the Abbott reference as loading classes from anywhere other than the classpath (Remarks: page 7, line 3 of the last paragraph).

Examiner respectfully disagrees: Examiner maintains that Abbott discloses such a limitation as "a stream source class loader", particularly loading the classes from a directory/hierarchy (*stream sources*) via Classpath. Abbott reference teaches class loading using Class Loader (Figure 1, and column 2, lines 30-42). In the Figure 1, it shows an arrow pointed to Classpath from Class Loader; this is as a reference for searching data files from a directory/hierarchy (*stream sources*) via Classpath. Classpath in this manner is only the interface/connection between the Class Loader to data files. Classpath provides a path to a file such as h.jar, e.class, or f.jar (See Figure 1 and column 3, lines 4-8, "other data files used by Java code) to load into Class Loader.

Since the Class Loader is implemented in JVM (Java Virtual Machine) (Figure 1). It is known that Java Virtual machine translates Java Bytecode into the native code for execution. The Java code such as h.jar and f.jar would be Java bytecode when they enter the JVM. Therefore the mechanism shown in Figure 1 reads the claimed limitations, including limitation, *a plurality of streaming sources* (hierarchy/directory containing h.jar, e.class, f.jar, etc.) *containing information including the location of data* (Figure 1 as the hierarchy/directory containing h.jar, e.class, f.jar (*streaming sources*)); *wherein requests for data* (h.jar or e.class, or f.jar) *are communicated from the stream source class loader* (Class Loader) *to the streaming sources* (hierarchy/directory) *via the interface* ('Classpath') *and, data passes from the stream sources to the stream source class loader via the interface, the streaming sources searching the data locations for the requested data* ('h.jar' or 'e.class', or 'f.jar', or 'a.class', etc. See column 2, lines 59-66).

Applicants further argue that the Abbott reference fails to disclose "*an interface coupled to the stream source class loader*" where Applicants pointed out "interface" in referring to "*requests for data are communicated from the stream source class loader to the streaming sources via the interface*".

Applicants' argument would be summarized as that the path's of Abbott merely denotes the search direction that is differed for reading byte data from sources and delivering it to the stream source class loader (Remarks: Page 8, lines 11-19).

Examiner respectfully responds: Since Claim 1 recites broadly, *an interface coupled to the stream source class loader*, the referencing 'path' from Class Loader, and Classpath shown in Abbott's Figure 1 meets means of "interface". Furthermore, the mechanism of Figure 1 clearly shows as it reads byte data from sources in the directory/hierarchy and delivers it to the class loader. The data is clearly loaded into class loader via the Classpath. If the data cannot be requested and read/delivered to the Class Loader (stream source class loader), then how can the file such as h.jar enter the JVM into the Class Loader? Moreover, Claim 1 also merely recites "*data passes from the stream sources to the stream source class loader via the interface*" shows no difference to the mechanism, Classpath, referencing arrows, as shown in Figure 1.

Applicants hold the same arguments of Claim 4 and Claim 8 as referring to limitations as described in claim 1 (Remarks: Page 9). Accordingly, the Examiner maintains that the same points as discussed in Claim 1 above also applied herein.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-12 are rejected under 35 U.S.C. 102(e) as being anticipated by Abbott, (US No. 6,542,887).

Given the broadest reasonable interpretation of followed claims in light of the specification.

As per claim 1:

Abbott discloses,

"A system, comprising:

a stream source class loader (see the figure (fig.1), 'Class Loader' and see column 3, lines 9-13, 'conventional native code library load mechanism') *retrieving streaming data* (h.jar, e.class, or f.jar) *to create a desired class object* (column 2, line 30 'responsive to a request to instantiate a class'); *an interface coupled to the stream source class loader* ('Classpath', and referencing 'Arrows'), *and a plurality of streaming sources* (Hierarchy/directory with the root h.jar referenced by Classpath) *containing information including the location of data* (see in the figure 1: referring to the arrows pointing to the class file directory and cache, and see column 2, lines 62-66), *wherein requests for data are*

communicated from the stream source class loader to the streaming sources (referencing arrows from Class Loader to hierarchy/directory) via the interface (Classpath) and, data passes from the stream sources to the stream source class loader via the interface, the streaming sources searching the data locations for the requested data (see figure 1, referring to the arrows from 'Classpath' to 'h.jar').

As per claim 2: Abbott discloses, "*The system according to claim 1, wherein the class object is a Java class object*" (column 3, line 8, 'Java code').

As per claim 3: Abbott discloses, "*The system according to claim 1, wherein the plurality of streaming sources includes one of a zip file stream source, a database stream source and a uniform resource locator stream source*" (see figure 1, '...zip', '...dll', '...jar').

As per claim 4: Abbott discloses, "*A class loader, comprising:*

a receiving module to receive a request for a desired class object (see in the figure 1, 'Class loader');

a stream source module containing information including the location of data which may be streamed to the class loader (see in the figure 1, Classpath and see column 2, lines 58-67, 'A class path is defined to point a list of locations where class files are found');

an interface module which receives requests for data from the receiving module (See in the figure 1, referring to the path between Class Loader and 'Classpath') and retrieves the streaming data from the stream source module (See in figure 1, referring to the path connected from 'Classpath' to class files directories); wherein the stream source module searches the data locations for the requested data (see column 2, lines 58-67, 'A class path is defined to point a list of locations where class files are found'); and *an instantiating module to receive the streaming data from the interface module and instantiate the class object (column 2, line 30 'responsive to a request to instantiate a class').*

As per claim 5: Abbott discloses, "*The class loader according to claim 4, wherein the stream source module includes a streaming source set by an application program*" (see column 3, lines 22-28, 'This means that an application can make call').

As per claim 6: Abbott discloses, "*The class loader according to claim 4, wherein the stream source module includes a plurality of streaming sources ('Classpath' and 'Libpath') wherein the stream source*

Art Unit: 2122

module searches each of the streaming sources to locate the desired data” (see column 2 62-65, referring to sub-directory’).

As per claim 7: Abbott discloses, *“The class loader according to claim 6, wherein the stream source module searches each of the streaming sources beginning with a most likely streaming source and proceeding sequentially in descending order (inherently in directory structure) of likelihood through the remaining streaming sources” (see column 2, lines 62-66, ‘sub-directory’).*

As per claim 8: Claim 8 is a method claim that has functionality corresponding to the functionality of claim 1. Claim 8 is rejected in the same reason set forth in connecting to the rejection of claim 1.

As per claim 9: Abbott further discloses the method, *“The method according to claim 8, further comprising the step of: instantiating the class object using the streamed data” (See column 2, lines 30-31, ‘instantiate a class’; and see column 3, lines 4-8, ‘load property files and other files used by Java code’ (Examiner note: Java code has means of streamed data)).*

As per claim 10: Abbott further discloses the method, *“The method according to claim 8, further comprising the step of: configuring the stream source including the location of the data for the class Object” (see column 3, lines 9-13, “the conventional native code mechanism...is augmented to search the classpath for the required library, and see column 2, lines 58-67, ‘A class path is defined to point a list of locations where class files are found’).*

As per claim 11: Claim 11 is a method claim that has functionality corresponding to the functionality of claim 3. Claim 11 is rejected in the same reason set forth in connecting to the rejection of claim 3.

As per claim 12: Abbott’s class loader is class of Java object-oriented programming. Common knowledge shows a class that consists at least a method. When being executed by a computer, it always returns a result. In the search for a native code library, Abbott assumes a search path is correct (column 3, lines 29-30, ‘file names need to be correctly set’, line 35, ‘the correct file names’) and the requested native code library must exist (column 3, lines 45-46, ‘If such a DLL is found’). This suggests that an error would be returned from the class loader if the requested native code library were not in the directory. This suggests the teaching of claim limitation: *“The method of claim 8, wherein, if the data associated with the class object is not found, an error is returned”.*

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

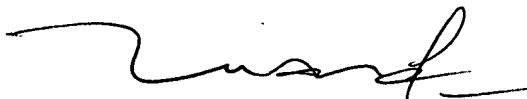
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (703) 308-9049. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam, can be reached on (703) 305-4552.

The fax phone numbers:

(703) 872-9306 (for formal communication intended for entry);

(703) 746-5429 (for informal or draft communication, please label "PROPOSED" or "DRAFT").

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.



TUAN DAM
SUPERVISORY PATENT EXAMINER

TTV
Patent Examiner
Art Unit: 2122
June 22, 2004